

REMARKS

In the Office Action dated November 27, 2007, claims 1-28 were pending. Claims 15-21 were rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 1-28 were rejected under 35 U.S.C. 102(e) as being anticipated by Soon et al. (hereafter, "*Soon*"), U.S. Pub. No. 2004/0001443.

Claims 15-21 were rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Applicants respectfully traverse this rejection. These are typical *in re Beauregard* claims, and as such, claim a physical apparatus (computer readable medium carrying instructions executable by a processor for executing the method in claims 1-7). The examiner agreed on the telephone February 21, 2008 that these claims are statutory. Applicants therefore submit that this rejection is improper and request that it be withdrawn.

Claims 1-28 were rejected under 35 U.S.C. 102(e) as being anticipated by Soon. Applicants respectfully traverse this rejection.

Furthermore, the present invention and Soon differ in their operation at a fundamental level. Soon appears to operate primarily on captured protocol data units (PDU). Soon also allows the editing of a captured PDU, or indeed, apparently, creation of a PDU from scratch ([0031]-[0035]). Then, a playout manager 36 is used to play back captured and/or modified PDUs from the PDU capture file ([0036]). One weakness with Soon, evident in view of the present invention, is that capturing and replaying PDUs, while flexible, is expensive in terms of resource consumption, and, thus, potentially not able to keep up with higher speed communications.

Contrast this with the present invention, where a protocol layer is primarily compiled code, as with protocol layers in actual operation. Then, a request to modify the protocol is displayed. In response, a command to modify the protocol is received, preferably in interpreted code, which allows it to be directly inserted into the protocol layer engine ([0031]). If the command is not in interpreted code, it is compiled into translated code ([0032]). As a result of this, the compilation is completed without having to recompile the code of the entire protocol. This allows a test to continue running during the insertion, since the entire protocol does not need to be recompiled ([0032]). The entire invention, being code based, is thus significantly more economical in terms of resource usage, and in many cases is likely more responsive to high speed communications.

Furthermore, as to claims 1, 8, 15, and 22, Applicants dispute that Soon provides for receiving commands to modify a protocol. Soon utilizes a sequence editor to modify test scenarios ([0023]; [0031]-[0034]). Nevertheless, the claims have been amended to make clear that the command to modify protocols comprises code. There is no suggestion, mention, or teaching in Soon that commands to modify a protocol involve, contain, or comprise code in any form, but the reference does contain significant suggestion that they involve editing PDU sequences instead. Thus, this claimed element is missing from the Soon reference. The remainder of the claims are dependent upon these independent claims, and should be allowable for similar reasons.

Furthermore, as to claims 2, 9, 16, and 23, Soon does not mention, suggest, or teach compiled or interpreted code. Compiled code is code that can be directly executed

by a processor, whereas interpreted code is code that can executed via an interpreter, or, typically, compiled into compiled code for direct execution ([0031]-[0035]). Soon does not work with “*code*”, but rather, with decoded PDUs that are replayed for the protocol testing ([0036]). Thus, this claimed element is missing from the Soon reference.

Furthermore, as to claims 7, 14, 21, and 28, Soon discloses the use of a state machine, but there is no mention, suggestion, or teaching of modifying the state machine. Rather, in Soon, the state machine appears to be static. Thus, this claimed element is missing for from the Soon reference.

Applicants therefore respectfully assert that significant claimed elements are missing from the Soon reference, that a prima facie case of anticipation has not been made, that this rejection of these claims is improper, and request that it be withdrawn.

Applicants believe that the above-identified application is now in condition for allowance and such action is respectfully requested.

If the Examiner has any questions regarding this application or this response, the Examiner is requested to telephone the undersigned at 775-586-9500.

Respectfully submitted,
SIERRA PATENT GROUP, LTD.

Dated: February 26, 2008

/bruce e. hayden/

Bruce E. Hayden
Reg. No.: 35,539

Sierra Patent Group, Ltd.
1663 Hwy 395, Suite 201
Minden, NV 89423
(775) 586-9500